Computation of Minimum Absent Words and Minimum p-Absent Words

Jean Toulzac, Aloys Paupe

Abstract

One of the most significant breakthroughs in computational biology has been the discovery of CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats). This system functions as an adaptive immunity mechanism, employed by bacteria to defend against bacterial viruses (phages) and predatory plasmids. It shares a notable analogy with computer antiviruses – every DNA molecule entering a bacterium is compared with an "internal database" of known "malicious code," and if a match is found, the molecule is destroyed, similarly to how computer antivirus eliminates malicious code before execution. However, to evade detection, viruses and plasmids have developed sophisticated anti-defense systems and evolved to avoid certain patterns, such as words stored in their "internal databases". All these mechanisms are of a great interest for the biologists, and new mechanisms might be discovered from the study of those words that plasmids avoid. The project's goal is to identify minimal words absent in a major plasmid dataset, to provide insights into words potentially targeted by CRISPR-Cas across diverse bacteria.

In this paper, we present naive algorithms for computing MAWs and pMAWs, as well as an improved approach based on properties of k-mers to reduce the search space.

1 Minimum Absent Words (MAW)

1.1 Definition

- **Absent word:** Let x and s be two strings on the alphabet Σ. We say that x is an *absent word* of S if neither x nor its reverse complement occur as a substring of S.
- **Minimal absent word:** Let x be a string of length $|x| > 1$ and S be a string. We say that x is a minimal absent word of S if both the following conditions hold:
  1. x is an absent word of S;
  2. for every substring w of x such that $|w| < |x|$, it holds that w or its reverse complement is a substring of at least one element of S.

1.2 Goal

The goal is to implement one (or more) approaches to enumerate all minimal absent words in each DNA sequence $s \in S$ that are shorter than a user-specified length k_{max}, where S denotes a user-provided set of input sequences.

1.3 Naive Algorithm

We do not implement a purely naive algorithm; instead, we introduce a slight optimization. Our approach is based on the following property: if all k-mers of a word S are known, then any MAW of length $k+1$ must be of the form

$$\alpha\omega \quad \text{or} \quad \omega\alpha,$$

where:

- ω is a k-mer occurring in S,
- $\alpha \in \Sigma$.

The generated candidate is then tested for occurrence in S; if it does not occur, it is considered a MAW.

Thus, rather than generating all possible words of length $k+1$, we restrict candidate generation to extensions of existing k-mers by a single character from the alphabet, either on the left or on the right.

1.4 Our Idea

2 Minimum p-Absent Words (pMAW)

2.1 Definition

- **Minimum p-Absent Words:** Let x be a string of length $|x| > 1$ and S be a set of strings. We say that x is a minimal p-absent word of S if both the following conditions hold:
    1. x is an absent word of S for at least $p|S|$ sequences $s \in S$;
    2. for every substring w of x such that $|w| < |x|$, it holds that w is not a p-absent word of S.

2.2 Goal

The goal is to implement one (or more) approaches to enumerate all minimal p-absent words in each DNA sequence $s \in S$ that are shorter than a user-specified length k_{max}, where S denotes a user-provided set of input sequences.

2.3 Naive Algorithm

2.4 Our Idea

3 Conclusion

References

[1] F. Crochemore, G. Fici, R. Mercaş, and S. Varricchio, *On the size of the set of minimal absent words*, BMC Bioinformatics, 2009. Available at: `https://link.springer.com/article/10.1186/1471-2105-10-137`